

Sampsa Isohätälä

## **VIRTUAALITODELLISUUSTUKI OHJELMISTON ALKUPERÄISEN KÄYTTÖLIIT- TYMÄN RINNALLE**

3D Visualizer

# **VIRTUAALITODELLISUUSTUKI OHJELMISTON ALKUPERÄISEN KÄYTTÖLIIT- TYMÄN RINNALLE**

3D Visualizer

Sampsa Isohätälä  
Opinnäytetyö  
Kevät 2020  
Tietojenkäsittelyn tutkinto-ohjelma  
Oulun ammattikorkeakoulu

## TIIVISTELMÄ

Oulun ammattikorkeakoulu  
Tietojenkäsittelyn tutkinto-ohjelma

---

Tekijä: Sampsa Isohätälä

Opinnäytetyön nimi: Virtuaalitodellisuustuki ohjelmiston alkuperäisen käyttöliittymän rinnalle

Työn ohjaaja: Jouni Juntunen

Työn valmistumislukukausi ja -vuosi: Kevät 2020

Sivumäärä: 26

---

Tämä opinnäytetyö käsittelee virtuaalitodellisuustuen teknistä toteutusta 3D Visualizer-ohjelmistoon. 3D Visualizer on Keysight Technologies Oy:n kehittämä, mobiiliverkkojen mittaustulosten analysointiin tarkoitettu ohjelmisto. Sen käyttötarkoitus on pystyä tarkastelemaan mittaustuloksia kolmiulotteisessa ympäristössä, jolloin mittauksesta olisi helpompi löytää ongelmakohtia.

Työn teoriaosuudessa käydään läpi yleiset käsitteet virtuaalitodellisuudelle sekä perusperiaatteet liikkumiselle, ohjaukselle sekä optimoinnille. Myös haasteita ja ongelmia tarkastellaan sekä kuinka välttää tai minimoida niitä.

Toteutuksessa tarkastellaan tarvittavia muutoksia alkuperäiseen ohjelmistoon, jotta tuki ja kaikki tarvittava virtuaalitodellisuuden kannalta saadaan sisäistettyä mukaan. Myös virtuaalitodellisuushahmon toteutus käydään läpi, sekä mihin ratkaisuihin päädyttiin ja mistä syistä.

Opinnäytetyön pohdinnassa todetaan virtuaalitodellisuustuen onnistuneen sekä käydään läpi työkalujen toimivuutta. Myös 3D Visualizerin jatkoa avataan hieman ja käydään läpi mahdollisia parannuksia tulevaisuutta varten.

---

Asiasanat: Virtuaalitodellisuus, Unity, 5G

## ABSTRACT

Oulu University of Applied Sciences  
Degree Programme of Business Information Systems

---

Author(s): Sampsa Isohätälä

Title of thesis: Virtual reality support alongside the software's original interface

Supervisor(s): Jouni Juntunen

Term and year when the thesis was submitted: Spring 2020      Number of pages: 26

---

This thesis deals with the technical implementation of virtual reality support for 3D Visualizer software. 3D Visualizer is software developed by Keysight Technologies Oy for analyzing measurement results of mobile networks. Its purpose is to be able to view the results in three-dimensional environment, which would make it easier to find problem areas in the measurement.

The theoretical part of the thesis reviews the general concepts of virtual reality as well as the basic principles of motion, control and optimization. Challenges and problems are also viewed, as well as how to avoid or minimize them.

The implementation will consider the necessary changes to the original software to internalize support and everything needed for virtual reality. The implementation of the virtual reality character is reviewed, as well as how solutions were reached and for what reasons.

In the reflection of the thesis, it is stated that virtual reality support has been successful, and the functionality of the tools is reviewed. The sequel to 3D Visualizer will also be opened up and possible improvements for the future will be reviewed.

---

Keywords: Virtual reality, Unity, 5G



# SISÄLLYS

1	JOHDANTO .....	6
2	VIRTUAALITODELLISUUS .....	11
2.1	Virtuaaliympäristö ja käyttöliittymä .....	11
2.2	Virtuaalitodellisuuslasit .....	12
2.3	Liikkuminen .....	13
2.4	Ohjaaminen .....	13
2.5	Optimointi .....	14
2.6	Työkalut .....	15
3	TOTEUTUS .....	16
3.1	Käyttöönotto .....	16
3.2	Liikkuminen .....	17
3.3	Käyttöliittymä virtuaalitodellisuuteen .....	18
3.4	Käyttöliittymän käyttöönotto .....	20
3.5	Optimointi .....	20
4	POHDINTA .....	23
	LÄHTEET .....	25

# 1 JOHDANTO

Tämä opinnäyte tuo on tehty yhteistyössä Keysight Technologies Oy:n kanssa. Keysight Technologies Finlandin Oulun toimisto sijaitsee Teknologia kylässä ja se on erikoistunut mobiiliverkkojen mittauksessa käytettäviin laitteistoihin ja ohjelmistoihin.

Opinnäytetyön tavoite on esitellä virtuaalimallin tekninen toteutus, johon kuuluu virtuaalimallin tuen lisääminen Unity-projektiin, liikkuminen, ohjaus, käyttöliittymän luonti sekä optimointi. Tässä esityksessä käyttöliittymällä tarkoitetaan valikkoa, josta voidaan säädellä ja suodattaa mittauksia. Käyttöliittymä myös sisältää ohjaimet, joilla voidaan ohjata hahmoa.

Tekninen toteutus tehdään Keysight Technologiesin kaupalliseen 3D Visualizer -ohjelmistoon, jota käytetään mobiiliverkkojen mittaustulosten analysointiin. Sen käyttötarkoitus on pystyä tarkastelemaan mittaustuloksia 3D-ympäristössä, jolloin mittauksista olisi helpompaa löytää ongelmakohtia, esimerkiksi rakennuksien ja kasvillisuuden estämiä signaaleja (katso kuva 1).



KUVA 1. 3D Visualizerin käyttöliittymä sekä mittausanalyysi

3D Visualizer on kehitetty työkaluksi mobiiliverkkoja rakentaville yrityksille helpottamaan verkkojen asennuksia ja säätämistä. Se toimii lisäosana Keysightin aikaisemmille tuotteille, Analyselle sekä Windcatcherille. Ne ovat tehokkaita ja täysin skaalautuvia analysointityökaluja vertailuanalyysille,

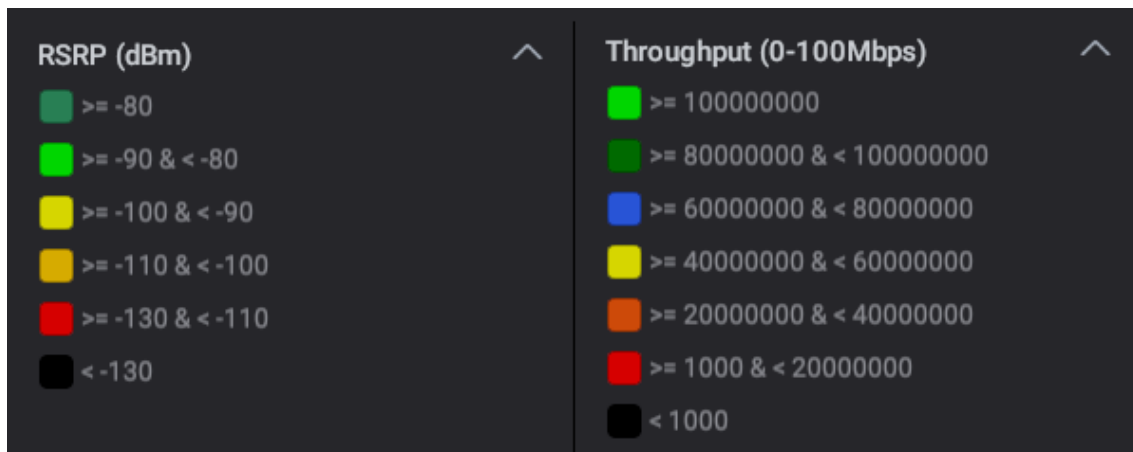
automatisoiduille vianetsinnälle sekä tilastollisille raporteille. Analysoituaan mittaustulokset toisessa näistä ohjelmista, voidaan ne avata 3D Visualizerissa tarkempaa analysointia varten. Ohjelmisto lähettää tarvittavan datan, jonka jälkeen 3D Visualizer piirtää sen kolmiulotteisesti koordinaattitietojen perusteella 2D-kartan päälle.

3D Visualizerissa on geneerinen ohjelmointirajapinta, joka pystyy vastaanottamaan kaiken sille lähetetyn datan ja näin ollen on yhdistettävissä Keysightin muihinkin tuotteisiin. Data voi sisältää useita parametreja, useilla suodattimilla. Ohjelmisto piirtää aluksi mittauspisteet 2D-kartan päälle, jonka jälkeen käyttäjällä on mahdollista ladata itsekuvattu 3D-malli tai geneerisesti luoda 2D-kartasta OSM Worldin avulla 3D-malli (katso kuva 2). Käyttäjällä on myös mahdollista hienosäätää 2D- ja 3D-karttoja, jos koordinaattipisteet eivät täsmää.



KUVA 2. Sama mittaus, samasta paikasta 2D-kartalla sekä molemmilla 3D-kartoilla

Käyttäjä pystyy suodattamaan ja säätämään mittaustuloksia tarpeidensa mukaan ja havaitsemaan mahdolliset ongelmakohdat. 3D Visualizer pystyy myös vastaan ottamaan nbf-tiedostoja, joista se luo tukiasemat sekä piirtää mahdolliset säteet, jos nämä tiedot löytyvät tiedostosta. Mittauspisteiden ja tukiaseman välille 3D Visualizer piirtää viivat, jos mittaustuloksen ja tukiaseman tiedot ovat yhtenäiset ja värittää viivat mittauspisteiden mukaan. Mittapisteiden värityksessä käytetään Keysightin luomaa väriselostetta, joka helpottaa mittauksen vahvuuden havainnointia (katso kuva 3).



KUVA 3. Väriseloste, kahdelle eri parametrille

3D Visualizeriin on luotu myös tuki tarkastella rakennuksien sisäisiä mittauksia. Käyttäjällä on mahdollista ladata ohjelmistoon rakennuksen pohjapiirustukset, koota niistä rakennus ja asettaa se mittauksessa sille tarkoitettuun paikkaan.

Toteutettavan virtuaalitodellisuustuen tulee mahdollistaa mittausanalyysien tarkastelun, suodattamisen ja hienosäätämisen virtuaalitodellisuuslaseilla. Ohjelmistoon tullaan luomaan kaksi erillistä ohjattavaa hahmoa, tietokone- sekä virtuaalitodellisuushahmo, joita ohjataan vuorotellen samassa ympäristössä mihin mittauks tulokset on analysoitu. Molemmilla hahmoilla tulee olemaan omat käyttöliittymät, jotka vaikuttavat samaan mittausanalyysiin. Käyttäjällä tulee olla mahdollisuus vaihdella näkymää näiden välillä. Tietokonehahmolla tarkoitetaan olemassa olevaa tapaa käyttää 3D Visualizeria tietokoneen avulla sekä sen käyttöliittymää. Virtuaalitodellisuudessa käyttäjällä tulee olla mahdollisuus liikkua vapaasti ja tarkastella tuloksia niin kaduntasolta kuin taivaasta. Käyttäjä voisi olla ikään kuin mittauspaikalla ja nähdä näkymättömät signaalit ja niiden vahvuudet.

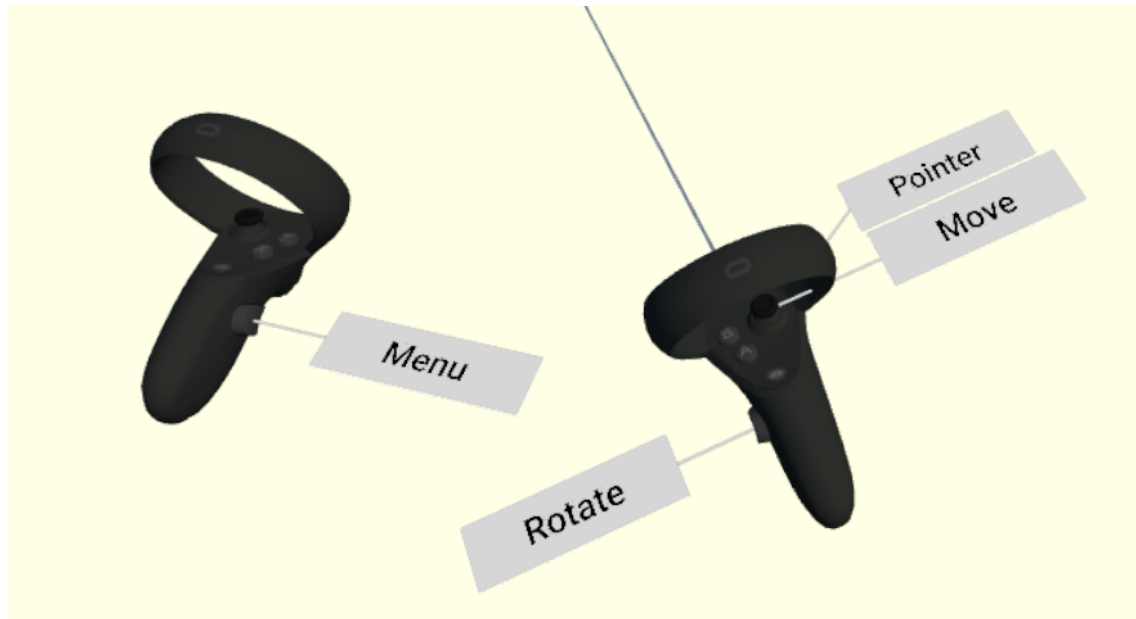
Alustavasti virtuaalitodellisuustuki luodaan ainoastaan Oculusin virtuaalitodellisuuslaseille, Riftille sekä Questille, Link-ohjelmiston kautta. Tässä toteutuksessa ohjelmiston tulisi automaattisesti tunnistaa virtuaalitodellisuuslasien ollessa kytkettynä tietokoneeseen ja luoda virtuaalitodellisuushahmo sekä käyttöliittymä. Virtuaalitodellisuustuki halutaan toteuttaa siten, että ohjelman käynnistyttyä käyttäjällä on mahdollisuus vaihdella näkymää tietokoneen ja virtuaalitodellisuuden välillä. Virtuaalitodellisuudessa hahmolla tulee olla mahdollisuus liikkua vapaasti, suodattaa ja hienosäätää mittauksia. Tietokone- sekä virtuaalitodellisuushahmon käyttöliittymien pitää myös olla yhtey-

dessä toisiinsa. Nykyisistä toiminnoista rakennuksien ja 3D-mallien luonti ja poistaminen sekä datan tuonti asetukset halutaan jättää pois. Näiden toimintojen tuonti virtuaalitodellisuuteen olisi jokseenkin epäkäytännöllistä, eikä niistä saataisi tarvittavaa hyötyä irti.

Koska tähän asti käyttöliittymä on luotu geneerisesti, on sen tekeminen toisen kerran suhteellisen yksinkertaista. Suunnitelma on luoda virtuaalitodellisuuskäyttäjän toiseen ohjaimeen tabletin tapainen objekti, virtuaalitodellisuushahmon käyttöliittymän.

Koska 3D Visualizerissa on jo olemassa käyttöliittymä tietokonehahmolle, on tärkeää, että virtuaalitodellisuuteen luotava käyttöliittymä vastaa alkuperäistä. Tämä helpottaa käyttöliittymän luontia virtuaalitodellisuuteen sekä käyttäjä voi helposti ja nopeasti sisäistää käyttöliittymän virtuaalitodellisuushahmolla.

Virtuaalitodellisuudessa ei ole käytössä hiirtä eikä näppäimistöä. Tämän takia joudutaan luomaan uusi tapa käyttäjälle olla vuorovaikutuksessa uuden käyttöliittymän kanssa. Tarkoitus on toteuttaa se oikean käden ohjaimesta lähtevän laserin avulla, joka toimii ikään kuin hiirenä. Oculuksen Touch ohjaimet sisältävät yhteensä viisi painiketta sekä yhden joystickin. Tämä antaa varsin hyvin mahdollisuuksia toteuttaa eritoimintoja eri painikkeiden avulla. Ajatus on käyttää oikean käden ohjainta pääohjaimena ja vasenta avustavana. Myöhemmin olisi mahdollisuus kysyä käyttäjän kätisyyttä ja vaihtaa ohjaimien roolia. Käyttäjäkokemusta helpotetaan luomalla sille avustusjärjestelmän, mikä pienille mutta helposti havaittavilla opasteilla kertoo mitä mistäkin painikkeesta tapahtuu (katso kuva 4). Näitä myös tullaan aktivoimaan ja deaktivoimaan, riippuen mikä toiminto on päällä.



KUVA 4. Oculus Touch ohjaimet virtuaalitodellisuudessa avustusjärjestelmän kanssa

## 2 VIRTUAALITODELLISUUS

Virtuaalitodellisuudella tarkoitetaan tietokoneen luomaa keinotekoista ympäristöä. Virtuaalitodellisuus voi luoda käyttäjälle kokemuksia, joita ei oikeasti ole olemassa tai simuloida jotain todellista. Käyttötarkoituksia virtuaalitodellisuudelle on rajattomasti, sitä voidaan käyttää muun muassa simulointiin, koulutukseen, ajanvietteeksi tai vaikkapa etäohjaukseen lääketieteellisissä operaatioissa. (Britta O'Boyle & Adrian Willings 2019, viitattu: 5.5.2020)

### 2.1 Virtuaaliympäristö ja käyttöliittymä

Virtuaaliympäristö on kolmiulotteinen, tietokoneen luoma tila, jonka asteikkona toimii X, Y sekä Z-akseli. Tilan lisäksi virtuaaliympäristö koostuu sinne sijoitetuista 3D-malleista ja mahdollisista toimintoista, kuten animaatioista tai koodatuista fyysisistä ominaisuuksista. 3D-mallit koostuvat verkkeistä ja niille määrätyistä tekstuureista. Virtuaaliympäristö vaatii myös hahmon sekä käyttöliittymän, jolla voidaan havaita, sekä olla vuorovaikutuksissa ympäristön kanssa. (Aviram Eisenberg 2020, viitattu 15.5.2020)

Virtuaalitodellisuuden ollessa täysin kolmiulotteinen, niin tulisi olla myös käyttöliittymän. Huomattavia etuja käyttöliittymän suunnittelussa virtuaalitodellisuuteen on näytön koko. Käyttöliittymä ei siis ole vain tietokoneen näyttöä suurempi, vaan teoriassa yhtä suuri kuin käyttäjän näkökenttä joka suuntaan. 2D-käyttöliittymä elementtien kiinnittäminen suoraan käyttäjän näkökenttään, on huono idea. Sen on todettu lisäävän matkapahoinvointia, eivätkä ne ole helposti luettavissa. Yleensäkin virtuaalitodellisuuden käyttöliittymässä tulisi aina välttää staattisia 2D-elementtejä. Yleisiä tapoja käyttöliittymän esittämiselle on sen asettaminen käyttäjän eteen leijuvana tauluna tai tekemällä siitä kädessä pidettävä. (Paul Mealy 2018, viitattu 11.5.2020)

Virtuaalitodellisuudessa tulisi välttää myös suuria määriä tekstiä sekä pientä fonttia. Suurella tekstillä varustetut lyhyet tekstikatkelmat ovat suositellumpia. Myös käyttäjän on vaikea syöttää pitkiä määriä tekstiä. Tehokas tapa tekstin syötölle virtuaalitodellisuudessa on vielä ratkaisematta globaalilla tasolla. Jos ohjelmisto vaatii tekstin syöttämistä, sen tapaa kannattaa harkita tarkkaan. (Paul Mealy 2018, viitattu 12.5.2020)

## 2.2 Virtuaalitodellisuuslasit

Virtuaalitodellisuuslaseilla tarkoitetaan päähän asetettavia laseja, joilla voidaan tarkastella virtuaalitodellisuutta sekä käsiin tulevia ohjaimia, joilla voidaan olla vuorovaikutuksessa virtuaalimaailman kanssa. Laseissa on molemmille silmille omat näytöt, joita katsotaan laseissa olevien linssien läpi. Näyttöjen kuvat ovat hieman erilaisia, jotka luovat kolmiulotteisen näkymän. Samalla tavalla kuten oikeassakin elämässä silmät näkevät hieman erikulmista asiat ja aivot yhdistävät näistä muodostavan kuvan. (Sourabh Purwar 2019, viitattu 11.5.2020)

Nykyisin virtuaalitodellisuuslaseilla on kaksi tapaa seurata liikettä. Ensimmäinen ja vanhempi tapa on lasien mukana tulevat majakat, jolla tarkoitetaan huoneeseen asetettavia kameroita seuraamaan käyttäjän liikkeitä. Toinen, uudempi tapa, on laseihin integroidut kamerat, jotka etsivät huoneesta kiintopisteitä ja näitä seuraamalla ohjelmisto pystyy laskemaan virtuaalitodellisuuslasien position ja rotaation. Ohjaimien seuraamiseen käytetään myös majakoita tai virtuaalitodellisuuslasien kameroita. On huomioitavaa, että ohjaimia pystytään ainoastaan seuraamaan, jos ne ovat kameroiden näköpiirissä. Tätä varten esimerkiksi Oculus Touch ohjaimiin on lisätty kiihtyvyysanturit, joilla pystytään suurin piirtein arviomaan, missä ohjaimet ovat, niiden ollessa piilossa kameroilta. Oculus Questista löytyy beta versiona käsien jäljitys, missä virtuaalitodellisuuslasien kameroiden avulla seurataan suoraan käyttäjän käsiä. (Sean Endicott 2019, viitattu 11.5.2020)

Menestyneimmät virtuaalitodellisuuslasien valmistajat ovat Sony, HTC ja Oculus, jotka ovat suunnitelleet virtuaalitodellisuuslasinsa pääosin pelaamiseen ja suurelle yleisölle. Suomalainen virtuaalitodellisuuslasien valmistaja Varjo, kehittää tämän hetken parhaimmat virtuaalitodellisuuslasit ja ne on suunnattu pääasiassa instituutioille simulaatio käyttöön. Tällaista käyttöä voivat olla esimerkiksi hävittäjälentäjien kouluttaminen. On myös syytä mainita, että suurin osa virtuaalitodellisuuslaseista on tarkoitettu käytettäväksi tietokoneen kanssa, jolloin tietokone tekee kaiken laskemisen. Tällöin virtuaalilasien tehtävä on ainoastaan näyttää laseissa tietokoneen lähettämät kuvat ja lähettää vastavuoroisesti tietokoneelle, sen hetkisen position, rotaation ja käyttäjän antamat syötteet. Tässä tapauksessa lasien tehokkuuden määrittää tietokoneen laskentateho. Oculuksen kehittämä Quest on itsenäinen järjestelmä, jossa on oma prosessori ja Android-käyttöjärjestelmä. Tällöin ohjelmistot ovat ladattavissa suoraan virtuaalitodellisuuslaseihin ja laskeminen tapahtuu laseissa. Tämä on erittäin rajoittava tekijä raskaiden ohjelmistojen suorituksessa, sillä virtuaalitodellisuuden sulava pyörittäminen on jo itsessään raskas operaatio. (Joe Bardi 2019, viitattu 11.5.2020; Sean Endicott 2019, viitattu 11.5.2020)



## 2.3 Liikkuminen

Yksi virtuaalitodellisuuden vahvuuksia on kyky luoda erittäin houkuttelevan näköisiä ympäristöjä, joita tutkia. Niistä ei tosin ole paljon hyötyä, jos niissä ei pysty liikkumaan. Liikkumiseen virtuaalitodellisuudessa on yleisesti käytössä kaksi tapaa. Teleporttaaminen, jossa käyttäjä siirretään suoraan haluttuun paikkaan. Se on yksinkertainen tapa liikkua ja näin saadaan samalla minimoitua käyttäjän kokema matkapahoinvointi (motion sickness). Toinen tapa on liikuttaa käyttäjää tasaisesti tiettyyn suuntaan, mikä on luonnollisempi tapa, mutta aiheuttaa tietyille käyttäjille erittäin vahvaa matkapahoinvointia. Toinen variaatio tästä on pitää käyttäjä paikallaan ja liikuttaa ympäristöä suhteessa häneen. Tämä luo illuusion, että käyttäjä itse liikkuisi. Ohjelmiston tulisi myös aina välttää käyttäjän liikuttamista, vaan kaiken liikkeen tulisi tapahtua vasta käyttäjän tehtyä liikkumistoiminto. (Paul Mealy 2018, viitattu 5.5.2020)

Matkapahoinvointi virtuaalitodellisuudessa johtuu yleensä käyttäjän liikuttamisesta. Käyttäjä näkee liikkeen muttei tunne sitä. Toinen tunnistettava huononolon aiheuttaja on viive, esimerkiksi pään kääntämisen ja kuvan renderöinnin välillä. Yleisimmät tavat ehkäistä matkapahoinvointia on, pitää virkistystaajuus tasaisena ja yhtenäisenä virtuaalitodellisuuslasien kanssa, päivittää pään positio ja rotaatio joka framella sekä välttää kiihtyvää liikettä. Myös teleporttaamisen käyttäminen liikkumiseen estää liikkeen havainnointia, mikä minimoi matkapahoinvointia. Yhdysvalloissa Columbian yliopisto on kehittänyt tätä varten menetelmän, missä visuaalisesti havaittu liikkeen aikana, automaattisesti pienentämällä käyttäjän näkökenttää saadaan ehkäistyä pahimpia oireita. Tätä varten löytyy valmiita ratkaisuja, joita on tarkoitus hyödyntää tässä toteutuksessa. (Paul Mealy 2018, viitattu 5.5.2020; Holly Evarts 2016, viitattu 2.5.2020)

## 2.4 Ohjaaminen

Ohjaaminen virtuaalitodellisuudessa tapahtuu yleisesti virtuaalilasien ohjaimilla tai katsomalla virtuaalitodellisuuslaseilla tiettyyn paikkaan tai suuntaan. Kaikkien virtuaalitodellisuuslasien valmistajien ohjaimissa on jotain erilaisuuksia, mutta ne yleensä sisältävät erimäärän painikkeita, joystickkejä tai touchpadeja. Kaikki painikkeet, joystickit ja touchpadit ovat täysin ohjelmoitavissa, joten kehittäjällä on täysin vapaat kädet ohjelmoida ne vastaamaan kehitettävän ohjelmiston tarpeita. Painikkeet voivat myös olla joko digitaalisia tai analogisia. Digitaalinen painike antaa tiedon onko painike pohjassa vai ei, kun taas analoginen painike kertoo, kuinka paljon painiketta on painettu.

Joidenkin valmistajien ohjaimissa on myös etäisyysantureita, jotka tunnistavat kuinka kaukana käyttäjän sormi on painikkeesta. Tämä lisää mahdollisuuden seurata käyttäjän sormien liikettä vielä tarkempaa, jota voidaan esimerkiksi käyttää käsien animointiin virtuaalitodellisuudessa. Käytössä on myös tieto ohjaimen positiosta ja rotaatiosta, joita voidaan myös hyödyntää syötteenä. Esimerkiksi liikkuminen voidaan toteuttaa tarkkailemalla ohjaimen rotaatiota, joka kertoo halutun liikkumissuunnan sekä joystickin arvoa, joka kertoo, kuinka nopeaa liikutaan eteen tai taakse, joystickkiä vetämällä tai työntämällä. (Scott Stein 2019, viitattu 11.5.2020)

Syötteiden antaminen muuttuu virtuaalitodellisuudessa. Enää käyttäjällä ei ole käytettävissä hiirtä eikä näppäimistöä, joten sitä varten tulee kehittää jokin uusi tapa kertoa, mitä painiketta milloinkin osoitetaan tai painetaan. Tähän on olemassa paljon valmiita, jo valmiiksi koodattuja ratkaisuja. Yleisin, mikä tuntuu olevan käytössä lähes kaikissa virtuaalitodellisuus sovelluksissa, on luoda ohjaimen laser, joka osoittaa haluttuun suuntaan ja toimii näin hiirenä. Myös on mahdollista luoda mekaaninen käyttöliittymä, missä käyttäjä joutuu käyttämään ohjainta kätenään ja painamaan ikään kuin fyysisesti painikkeita. Tällä tavalla käyttöliittymästä voi saada viihdyttävän, mutta sen käytön tehokkuus kärsii ja on erittäin työläs luoda. (Jose Somolimos 2020, viitattu 11.5.2020)

## 2.5 Optimointi

Optimointi virtuaalitodellisuusohjelmistolle on kiinni mitä virtuaalitodellisuuslaseja käytetään ja millä ohjelmistolla virtuaalitodellisuutta rakennetaan. Haastavaa optimoinnista tekee se, että virtuaalitodellisuuslaseissa on kaksi näyttöä, yksi molemmille silmille. Tämä tarkoittaa teoriassa sitä, että kaikki pitää renderöidä kaksi kertaa. Pääidea kuitenkin optimoinnille virtuaalitodellisuuteen luotavissa ohjelmissa on käyttää tarkempia tekstuureja ja mahdollisimman vähän verteksejä objekteissa. Tämä helpottaa huomattavasti virtuaalitodellisuuden renderöintiä. Suurin vaikutus optimoinnissa on käytettävän ohjelmiston projekti asetukset. Tässä on hyvä lukea ja perehtyä käytettävän kirjaston suosituksiin. (Oculus 2020, viitattu 11.5.2020)

Myös virtuaalitodellisuuslasien virkistystaajuus vaikuttaa virtuaalitodellisuuden sulavuuteen. Tarkoituksena olisi saada ohjelmiston virkistystaajuus yhtenäistettyä lasien virkistystaajuuden kanssa. On myös huomioitavaa, että mitä tiheämpää päivitetään, sitä enemmän laskutehoa siihen kuluu. Sulavuus ja käyttömukavuus on korkeimpia prioriteetteja virtuaalitodellisuudessa. (IrisVR 2020, viitattu 11.5.2020)

## 2.6 Työkalut

Ensimmäinen asia, joka täytyy ottaa huomioon, kun valitaan oikeita työkaluja virtuaalitodellisuuden kehitykseen, on tietää mille virtuaalitodellisuuslaseille ohjelmisto halutaan tuottaa. Virtuaalitodellisuustekniikan ollessa vielä kesken, on sen kategorioiden rajat epäselviä. Näillä kaikilla on kuitenkin sama tavoite, luoda käyttäjälle syvällisiä kokemuksia. Tapahtui se sitten 360-asteen panoraama videoilla, täysin tietokoneen luomalla sisällöllä tai näiden molempien sekoituksella. Tämä on tärkeää sisäistää ennen, kun valitaan oikeaa työkalua virtuaalitodellisuusohjelman toteutukselle. Työkaluna voidaan käyttää pelimoottoreita tai ohjelmistokehityspaketteja (SDK, Software development kit). Työkalut sisältävät yleensä apuohjelmia, joista löytyy peruslogiikka, niin liikkumiselle sekä tavallaan olla vuorovaikutuksessa ympäristöön. (Wendy Wise 2017, viitattu 15.5.2020; Valeriy Ilchenko 2019, viitattu 15.5.2020)

Yksi suosituimmista työkaluista virtuaalitodellisuudelle on Unity-pelimoottori, jolla voidaan luoda sovelluksia monille eri alustoille. Sillä voidaan luoda niin 2D- ja 3D-sovelluksia sekä virtuaali- sekä lisättyä todellisuutta (Augmented reality). Unity Technologies julkaisi Unityn vuonna 2005. Unitystä on maksuton versio sekä maksullinen Unity Pro versio. Unityssa käytettävä koodauskieli on C#. (Samuel Axon 2016, viitattu 11.5.2020) Unity-pelimoottorilla virtuaalitodellisuus sovelluksien tekeminen on suhteellisen helppoa. Periaatteessa täytyy vain valita projektin asetuksista ”Virtual Reality Supported”, ja valita alusta, missä ohjelmiston tulisi toimia. Tässä vaiheessa projekti tukee virtuaalitodellisuutta. Seuraavaksi täytyy rakentaa hahmo ja sille tarvittava logiikka. (Unity 2020, viitattu 12.5.2020)

### 3 TOTEUTUS

Virtuaalitodellisuustuen toteutuksessa tullaan projektiin lisäämään uusi hahmo, eli käytössä tulee olemaan tietokonehahmo sekä virtuaalitodellisuushahmo. Hahmoista vain toinen tulee olemaan aktiivinen kerrallaan, ja näin saadaan pidettyä kaksi erillistä järjestelmää. Virtuaalitodellisuuden ollessa aktiivinen, tietokoneennäytöllä tulee myös näkyä mitä virtuaalitodellisuushahmo näkee.

#### 3.1 Käyttöönotto

Ensimmäinen asia mikä täytyy tehdä, on aktivoida Unityn projekti asetuksista tuki virtuaalitodellisuudelle. Koska toteutus tehdään aluksi vain Oculuksen virtuaalitodellisuuslaseille, projektiin täytyy tuoda Oculuksen virtuaalitodellisuusapuohjelmienlaajennuspaketti (OVR), mikä on ladattavissa ilmaiseksi Oculuksen verkkosivuilta. Ensiksi lisätään ympäristöön OVR-paketissa tulevan kamera-laite (camera rig), joka pitää sisällään peruslogiikan katsomiselle virtuaalitodellisuudessa. Kamera-laiteesta löytyy myös tyhjät objektit ohjaimille, jotka edustavat ohjaimien sijaintia virtuaalitodellisuudessa. Seuraavaksi annetaan 3D-mallit OVR paketista tyhjiille objekteille, ja näin voidaan seurata missä ohjaimet ovat suhteessa käyttäjään virtuaalitodellisuudessa.

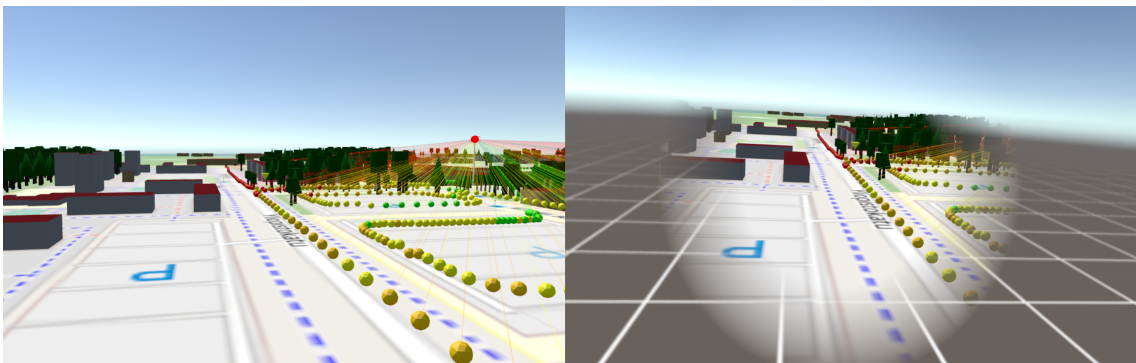
Seuraavaksi luodaan uusi skripti, jonka tehtävä on, tarkistaa ohjelmiston käynnistyttyä, onko virtuaalitodellisuuslasit kytkettynä tietokoneeseen, ja toimia sen mukaan. Jos laseja ei ole kytkettynä, mitään ei tapahdu. Jos lasit ovat kytkettynä, se kutsuu aktivointi metodia, mistä voidaan lähteä rakentamaan logiikkaa virtuaalitodellisuuskäyttäjälle.

Kun ohjelmisto tunnistaa virtuaalitodellisuuslasien ollessa kytkettynä, luomme nykyiseen käyttöliittymään VR-painikkeen, josta voidaan aktivoida virtuaalitodellisuushahmo. Tätä varten täytyy myös luoda toiminto, millä voidaan poistua virtuaalitodellisuudesta. Tämä kanssa päädyttiin ratkaisuun, jossa virtuaalitodellisuuden ollessa käytössä, emme kuuntele mitään muuta syötettä tietokoneen puolella kuin escape-painiketta. Tämä taas aktivoi tietokonehahmon ja sulkee virtuaalitodellisuuden.

## 3.2 Liikkuminen

Unityssa luotava ympäristö on teoriassa loputon. Tästä syystä liikkuminen ympäristössä kannattaa rajata jotenkin. Jos kyseessä olisi suljettu ympäristö, esimerkiksi huone, sen seinien käyttäminen alueen rajaamiseen on luontainen tapa rajata liikkuminen. 3D Visualizerin tapauksessa ympäristössä ei ole selviä rajoja joten, sen täytyy toteuttaa rajaaminen eri tavalla. Tässä toimisi esimerkiksi määrittämällä koodiin minimi- ja maksimikorkeus, joka estäisi menemisen visualisoidun maatasen alapuolelle tai lentämisen liian korkealle (Y-akseli). Liikettä horisontaalisti ei tulla rajaamaan (X ja Z-akseli).

Liikkumisessa päädyttiin ottamaan esimerkkiä jo valmiista ratkaisuksista, Google Earth, joka joiltain osin toimii samaan tyyliin kuin 3D Visualizer. Liikkumisesta ei haluta tehdä realistista vaan päädyttiin lentämiseen, missä käyttäjä osoittaa ohjainta haluttuun suuntaan ja säätelee nopeutta joystickilla. Tämä tapa aiheuttaa erityisen paljon matkapahoinvointia, joten käyttöön otetaan Sigtrapin luoma VR Tunnelling Pro-lisäosa. Lisäosa toimii niin, että aina liikkumisen aikana käyttäjän näkökenttää pienennetään, ja näkökentän ulkopuolelle jäävän tilan päälle piirretään ristikko. Ristikon tarkoitus on simuloida lattian tasoa, jonka päällä käyttäjä seisoo (katso kuva 5).



KUVA 5. Normaalinäkö sekä VR Tunnelling Pro-lisäosa aktivoituna

Liikkuminen tapahtuu oikean käden ohjaimella. Osoittamalla sillä haluttuun suuntaan, ja joko työntämällä tai vetämällä joystickkia, riippuen halutaanko liikkua eteen vai taakse. Tässä tarkastellaan oikean käden rotaatioita, josta saadaan haluttu suunta sekä joystickin vertikaalista akselia. Joystickin ollessa analoginen syöte, saadaan tieto kuinka paljon joystickkia on työnnetty mihinkin suuntaan. Tämä auttaa käyttäjää säätelemään nopeuttaan työntämällä joystickkia vain halutun verran.

Ympäristön ollessa geneerisesti luotu, voi se olla halkaisijaltaan useita kilometrejä. Tätä varten luodaan vielä toiminta, jossa käyttäjän liikkeessa, aktivoimme molempien ohjeimien liipaisin painikkeet. Vasemman käden liipaisin painikkeesta, voidaan liikkumisnopeus puolittaa. Oikean käden vastaavasta painikkeesta voidaan taas, moninkertaistaa nopeus. Mikä helpottaa pitempien etäisyyksien liikkumista.

Käyttäjän käyttäjäkokemuksen parantamiseksi, täytyy myös ottaa huomioon, että kaikki käyttäjät eivät halua käyttää ohjelmistoa seisaaltaan. Tässä ongelma on kääntyminen virtuaalitodellisuudessa. Ongelma ratkaistaan luomalla toiminta, missä käyttäjä voi kääntää hahmoa ohjaimen avulla. Tämä toteutetaan määrittämällä toiminta oikean käden puristuspainikkeeseen, jonka ollessa pohjassa hahmo kääntyy ohjaimen rotaation mukaan. Kun painike painetaan pohjaan, tallennetaan ohjaimen rotaatio. Painikkeen ollessa painettuna verrataan joka framella ohjaimen rotaatiota tallennettuun rotaatioon ja hahmoa pyöritetään sen mukaan mikä erotus tallennetun rotaation ja nykyisen rotaation välillä on. Tämän toiminnan aikana myös aktivoidaan VR Tunnelling -lisäosa, koettavan matkapahoinvoinnin ehkäisemiseksi.

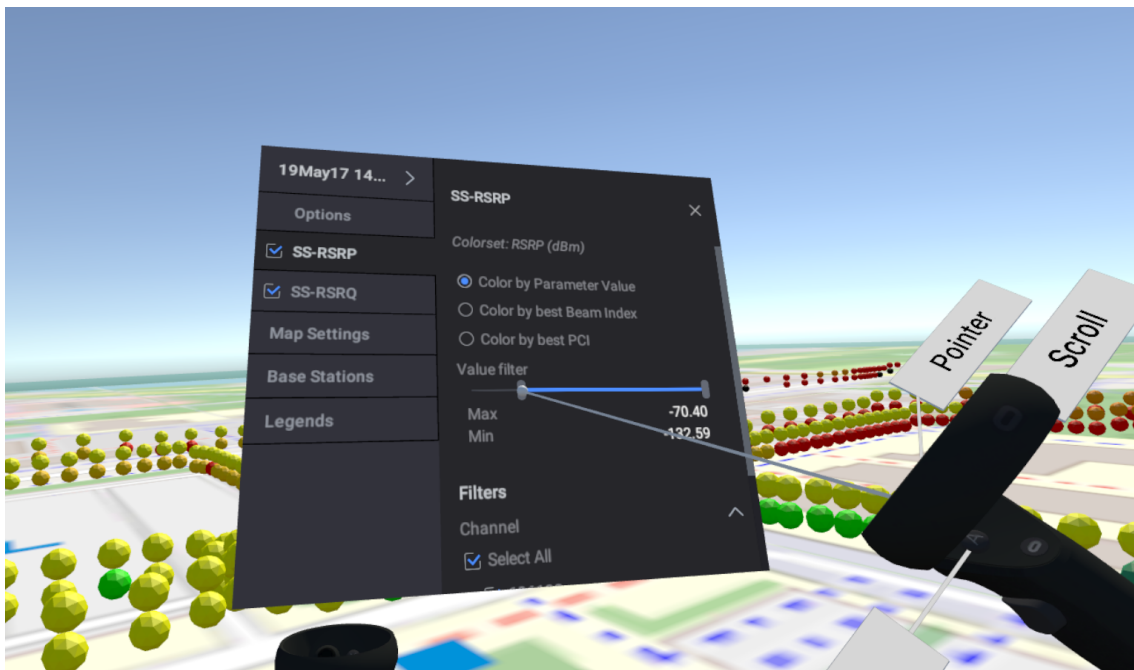
### **3.3 Käyttöliittymä virtuaalitodellisuuteen**

Koska käyttöliittymä luodaan jo geneerisesti tietokonehahmon puolella, virtuaalitodellisuuteen sen luominen käy suhteellisen yksinkertaisesti. Periaatteessa luodaan sama käyttöliittymä kaksi kertaa ja asetetaan kullekin hahmolle oma käyttöliittymä. Tämä ratkaistaan lisäämällä yksi boolean parametriseksi, kaikille käyttöliittymän luonti funktioille. Boolean kertoo vain käyttöliittymän luontilogiikalle, minkä vanhemman alle käyttöliittymä luodaan. Vasemman ohjaimen lapseksi luodaan tyhjä objekti, mihin virtuaalitodellisuushahmon käyttöliittymä luodaan. Luotuaan käyttöliittymän tietokonehahmolle, tarkitetaan jälleen, onko virtuaalitodellisuuslasit kytkettynä tietokoneeseen, jos on, luomme saman käyttöliittymän toisen kerran. Parametria tarkkailemalla luodaan myös ehdot, mitä kaikkea halutaan luoda virtuaalitodellisuuteen.

Käyttöliittymän luonnin jälkeen seuraava tehtävä on luoda syöttöjärjestelmä virtuaalitodellisuutta varten. Tässä Oculuksen luoma ratkaisu toimii hyvin. Unityssa syötteet hoitaa tapahtumajärjestelmä (Event System), jolle on määrätty syöttömoduuli (Input module). Alkuperäinen versio käyttää Unityn automaattisesti luomaa syöttömoduulia, Standalone Input Module skriptiä, mikä tunnistaa hiiren painallukset. Tämä ei kuitenkaan toimi enää virtuaalitodellisuudessa, koska käytössä ei ole

hiirtä. Oculuksen ratkaisu on luoda uusi syöttömoduuli, OVR Input Module skripti. tapahtumajärjestelmän tukiessa ainoastaan yhtä syöttömoduulia kerrallaan, joudutaan myös siihen luomaan loogiikka, joka tietyn käyttäjän ollessa aktiivinen, aktivoi sitä vastaavan syöttömoduulin. Tässä paras ratkaisu on lisätä alkuperäisen syöttömoduulin lisäksi OVR:n syöttömoduuli olemassa olevalle tapahtumajärjestelmän objektille. Vaihdettaessa hahmoa, aktivoidaan sitä vastaava syöttömoduuli ja deaktivoidaan toinen.

Tällä hetkellä syöttömoduuli toimii virtuaalitodellisuudessa siten, että oikean ohjaimen päästä lähtee laser, joka näyttää selvästi mihin käyttäjä osoittaa ohjaimella (katso kuva 6). Laserin tunnistettua tietty layer, tässä tapauksessa UI Layer, se aktivoi kursorin siinä kohdassa missä laser kohtaa layerin. Kursorina toimii pieni 3D-pallo. Valinta painikkeeksi valikoitui A painiki oikean käden ohjaimessa.



KUVA 6. Käyttöliittymä virtuaalitodellisuudessa

Virtuaalitodellisuushahmon käytön helpottamiseksi luodaan sille myös avustusjärjestelmä. Riippuen mikä toiminto käyttäjällä on menossa, opasteilla tullaan näyttämään, mitkä painikkeet ovat käytössä ja mitä niistä tapahtuu. Tämä helpottaa käyttöliittymän sisäistämistä ja nopeuttaa sen käyttöä. Joka painikkeelle tullaan luomaan oma avustuskyltti, jota voidaan kytkeä päälle ja pois sekä muuttamaan sen esittämää tekstiä. Tällä tavoin voidaan sitoa samoille painikkeille eri funktioita eri toimintojen ollessa päällä.

### 3.4 Käyttöliittymän käyttöönotto

Tässä vaiheessa käyttöliittymä toimii virtuaalitodellisuudessa, mutta käyttöliittymät eivät ole tietoisia toisistaan. Esimerkiksi jos suodatetaan jotain arvoja tietokonehahmolla, haluttu toimenpide tapahtuu, mutta kun siirrytään virtuaalitodellisuuteen, voidaan nähdä muutokset, mutta käyttöliittymä näyttää silti aloitusarvoja.

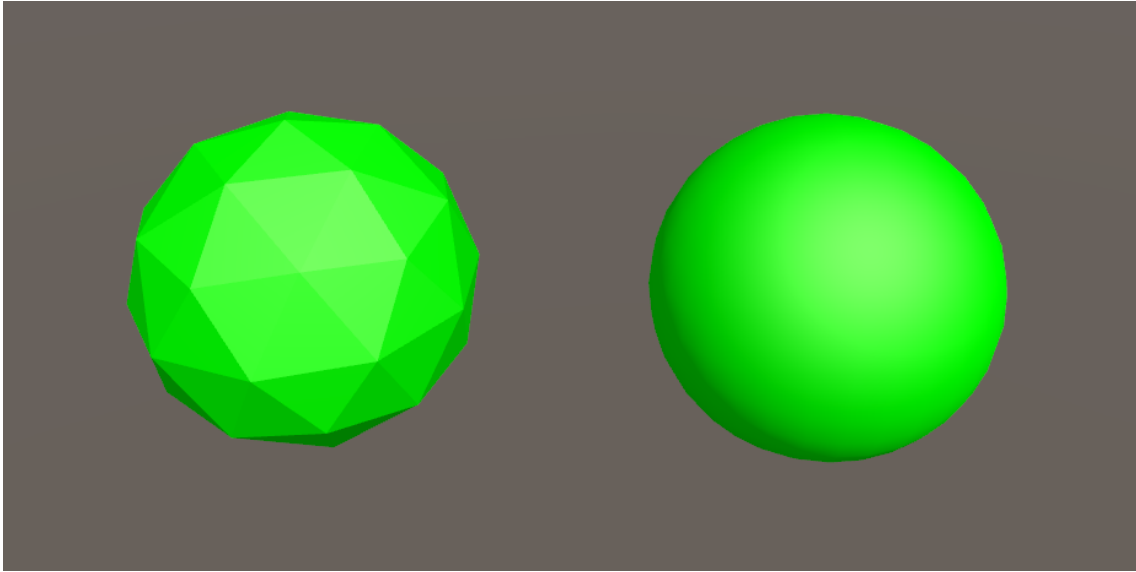
Käyttöliittymien yhdistäminen voidaan ratkaista muutamalla eri tavalla. Ensimmäinen tapa yhdistää virtuaalitodellisuuskäyttöliittymä suoraan alkuperäiseen. Esimerkiksi jos painetaan, piilota parametri – painiketta. Ei kutsuta haluttua metodia, vaan virtuaalisesti painetaan vastaavaa painiketta alkuperäisessä käyttöliittymässä. Toinen vaihtoehto, joka päättyy tähän toteutukseen, toimii itsenäisesti. Molemmilla käyttöliittymillä on viittaukset samoihin toimintoihin, mutta ovat edelleen erillään toisista. Synkronointi tapahtuu, kun vaihdetaan hahmoa. Siirryttäessä hahmosta toiseen, tarkistetaan kaikki mahdolliset painikkeet sekä liukukytkimet ja arvot kopioidaan niitä vastaaville objekteille aktivoitavan hahmon käyttöliittymässä.

Tämä myös tarkoittaa sitä että, kutsuttavien metodien alkuun täytyy tehdä tarkistus, jotta tiedettäisiin, kumman käyttöliittymän suodattimia pitäisi milloinkin käyttää. Tästä syystä luodaan kopiot muuttujista virtuaalitodellisuuden käyttöliittymälle, jotta ohjelmisto ymmärtää kumpi hahmo häntä kutsuu ja kumman hahmon suodattimia käyttää.

### 3.5 Optimointi

Koska alkuperäinen ohjelma on jo optioitu ja tehty ainoastaan tietokoneelle, täytyy nyt ottaa huomioon myös virtuaalitodellisuuden luomat vaatimukset. Pääidea virtuaalitodellisuuteen luotavissa ohjelmissa on käyttää parempia tekstuureja ja mahdollisimman vähän verteksejä. Tästä syystä vaihdetaan alkuperäiset mittapisteet, pyöreät pallot, lowpoly palloihin (katso kuva 7). Mikä laskee huomattavasti luotavien verteksien määrää.





*KUVA 7. Uusi ja vanha mittauspiste rinnakkain*

Muuten siirtyminen tietokoneversiosta virtuaaliodellisuuteen on lähinnä projekti asetusten muuttamista. Projektin laatu asetuksissa (Quality settings), tekstuuri laatu nostetaan maksimiin, aiemmin sanoman idean mukaan. Varjot on otettu täysin pois käytöstä. Varjojen luonti ja päivittäminen on raskas operaatio, eikä millään lailla oleellista käyttötarkoituksen takia. Oculusin OVR paketti suosittelee reunojen pehmentämisen (Anti-aliasing) nostamista nelinkertaiseksi. Projektin pelaaja asetuksista (Player settings) laitetaan Stereo Rendering Moden single passiksi. Nämä säädöt tulivat siis aikaisemmin olevien optimointien rinnalle.

Kehityksessä seurataan tiheästi Unityn tarjoamaa Profiler-työkalua, jota käytetään selvittämään mihin kaikkeen laskutehoa kuluu ohjelmiston pyöriessä. Tämä on helpottaa ohjelmiston optimointia. (Josh 2019, viitattu 11.5.2020)

Unityssa pystyy myös määrittämään mikä on ohjelmiston aika-askel (fixed timestep). Virtuaaliodellisuudessa aika-askel halutaan yhtenäistää lasien oman virkistystaajuuden (fps, frames per second) kanssa. Oculus Questin virkistystaajuuden ollessa 72 framea sekunnissa, halutaan se sama taajuus saada ohjelmiston puolelle, jotta kaikki liikkuisi synkronisesti virtuaaliodellisuuslasien kanssa. Virkistystaajuuden laskeminen on yksinkertaista, koska haluamme tietää minkä ajan välein päivitämme. Kaava on seuraava, 1 sekunti / tavoitevirkistystaajuus. Oculus Questin tapauksessa se on  $1 / 72 = 0.01288889$ . On myös huomioitavaa, että mitä tiheämpää päivitämme, sitä enemmän laskutehoa siihen kuluu, mutta kuten virtuaaliodellisuudessa yleensäkin, käyttömukavuus on yksi korkeimmista prioriteeteista.

Lisähaasteen tähän luo myös, että toiveena 3D Visualizerilla on näyttää tietokoneennäytöllä se, mitä käyttäjä näkee virtuaalitodellisuudessa. Mikä tarkoittaa, että meidän täytyy renderöidä yhteensä kolme kuvaa joka framella.

## 4 POHDINTA

Tämän opinnäytetyön tarkoitus oli toteuttaa virtuaalitodellisuustuki 3D Visualizer-ohjelmistoon. Työssä käytiin läpi yleisimmät käsitteet, jotka on hyvä ymmärtää ennen kehityksen aloittamista. Myös virtuaalitodellisuuden haasteita käsiteltiin ja kuinka niitä voidaan välttää tai minimoida.

Virtuaalitodellisuustuki on nyt valmis, ja se lähtee seuraavaksi testaukseen. Toteutukseen saatiin sisällettyä kaikki vaatimukset, jotka kirjattiin ennen projektin aloitusta. Virtuaalitodellisuustuki antaa nyt käyttäjälle mahdollisuuden olla ikään kuin paikan päällä. Myös mittasuhteet tulevat paremmin esille verrattuna alkuperäiseen toteutukseen. Olen tyytyväinen lopputulokseen, mutta esiin nousi joitakin ongelmakohtia, kuten käyttöliittymän käytettävyys sekä yhteysongelmat Oculus Link-ohjelmiston kanssa. Näiden toteutusta tulee vielä testailla ja kehittää.

Minulla oli hieman aikaisempaa kokemusta virtuaalitodellisuudesta ja sille luotavista sovelluksista. Opin kuitenkin toteutuksen aikana paljon lisää virtuaalitodellisuuden haasteista, kuinka ehkäistä tai välttää niitä. Erityisesti matkapahoinvoinnin ehkäisyyn kannattaa kiinnittää paljon huomiota ja miettiä tarkkaan mikä on paras ratkaisu kehitettävän ohjelmiston kannalta. Oli myös erittäin mielenkiintoista lähteä toteuttamaan tukea, olemassa olevan ohjelmiston rinnalle, siten että alustaa voi vaihdella kesken ohjelmiston suorituksen. Tähän olen erittäin tyytyväinen, ottaen huomioon, etten löytänyt vastaavaa olemassa olevaa ohjelmaa.

Unity-pelimoottori työkaluna toimi hyvin tässä toteutuksessa. Erityisesti Unityn dokumentoinnin sekä yhteisön ollessa laaja, aiheesta löytyy hyvin tietoa. Myös Sigtrapin luoma VR Tunnelling Pro-lisäosa ratkaisi pitkälti matkapahoinvointi ongelman. Sen ollessa ilmainen sekä suhteellisen helppo implementoida, olen ollut siihen erityisen tyytyväinen. Oculusin Link-ohjelmiston kanssa on joitain haasteita virtuaalitodellisuuslasien tunnistuksessa, mutta tähän on odotettavissa muutoksia, Link-ohjelmiston ollessa vielä beta-versio.

3D Visualizerin kehitys tulee jatkumaan Keysightilla. Tulevaisuuden näkymät toteutettavalle tuelle, ovat pitkälti kiinni millaista palautetta saadaan testauksesta. Tuki tulee vaatimaan vielä hienosäätöä liittyen liikkumiseen sekä käyttäjäkokemuksen parantamiseksi. Jatkossa pystymme luomaan toiminnon, joka käyttäjän kätisyyttä kysymällä pystyy vaihtaa ohjaimien toiminnot peilikuviksi, jotta ohjelmiston käyttö olisi helpompaa myös vasenkätisille. Tuen käyttöliittymä on myös mahdollista

jatkossa muuttaa, mikäli havaitaan, että nykyinen on liian vaikea käyttää. Yksi vaihtoehto tähän on luoda käyttöliittymästä leijuva valikko käyttäjän eteen, mikä estää käden tärinästä johtuvaa epätarkkuutta. Virtuaalitodellisuuden tukiessa tällä hetkellä ainoastaan Oculuksen virtuaalitodellisuuslaseja, olisi tarkoitus luoda tuki myös muiden valmistajien lasseille. Loppujen lopuksi jatkokehitys tulee olemaan kiinni asiakkaiden sekä markkinoiden vaatimuksista.

## LÄHTEET

Aviram Eisenberg 2020. VR Environment: What is it and how does it work? Viitattu 15.5.2020 <https://appeal-vr.com/blog/virtual-reality-environment-what-is-it-and-how-it-works/>.

Britta O'Boyle & Adrian Willings 2019. What's VR? Virtual reality explained. Viitattu 5.5.2020, <https://www.pocket-lint.com/ar-vr/news/136540-what-is-vr-virtual-reality-explained>.

Holly Evarts 2016. Fighting Virtual Reality Sickness. Viitattu 2.5.2020, <https://engineering.columbia.edu/news/fighting-virtual-reality-sickness>.

IrisVR 2020. The importance of frame rates. Viitattu 11.5.2020 <https://help.irisvr.com/hc/en-us/articles/215884547-The-Importance-of-Frame-Rates>.

Joe Bardi 2019. What is virtual reality. Viitattu 11.5.2020, <https://www.marxentlabs.com/what-is-virtual-reality/>.

Jose Somolimos 2020. How to design common/basic interactions in VR. Viitattu 11.5.2020 <https://www.accedo.tv/design-commonbasic-interactions-vr/>.

Josh 2019. How to debug and fix performance problems in Unity with the profiler tool. Viitattu 11.5.2020, <https://codingchronicles.com/unity-vr-development/how-to-debug-and-fix-performance-problems-in-unity-with-the-profiler-tool>.

Oculus 2020. Guidelines for VR performance optimization. Viitattu 11.5.2020, <https://developer.oculus.com/documentation/native/pc/dg-performance-guidelines/>.

Paul Mealy 2018. Best Practices and Virtual Reality Design Principles. Viitattu 5.5.2020, <https://www.dummies.com/software/best-practices-and-virtual-reality-design-principles/>.

Samuel Axon 2016. Unity at 10: For better and worse game development has never been easier. Viitattu 11.5.2020, <https://arstechnica.com/gaming/2016/09/unity-at-10-for-better-or-worse-game-development-has-never-been-easier/>.

Scott Stein 2019. Valve Index's new VR controllers feel like the future of gaming. Viitattu 11.5.2020, <https://www.cnet.com/news/valve-indexs-new-vr-controllers-feel-like-the-future-of-gaming/>.

Sean Endicott 2019. Oculus Quest: Everything you need to know! Viitattu 11.5.2020, <https://www.androidcentral.com/echo-vr-open-beta-brings-zero-gravity-multiplayer-your-oculus-quest>.

Sourabh Purwar 2019. Designing user experience for virtual reality applications. Viitattu 11.5.2020, <https://uxplanet.org/designing-user-experience-for-virtual-reality-vr-applications-fc8e4faadd96>.

Unity 2020. VR Overview. Viitattu 12.5.2020 <https://docs.unity3d.com/540/Documentation/Manual/VROverview.html>.

Valeriy Ilchenko 2019. Viitattu 15.5.2020 <https://www.byteant.com/blog/how-to-pick-the-best-tool-for-vr-development/>.

Wendy Wise 2017. Viitattu 15.5.2020 <https://www.oreilly.com/content/how-to-pick-the-right-authoring-tools-for-vr-and-ar/>.